Brief intro to

improving LLM usage performance

In particular, RAG

Contents

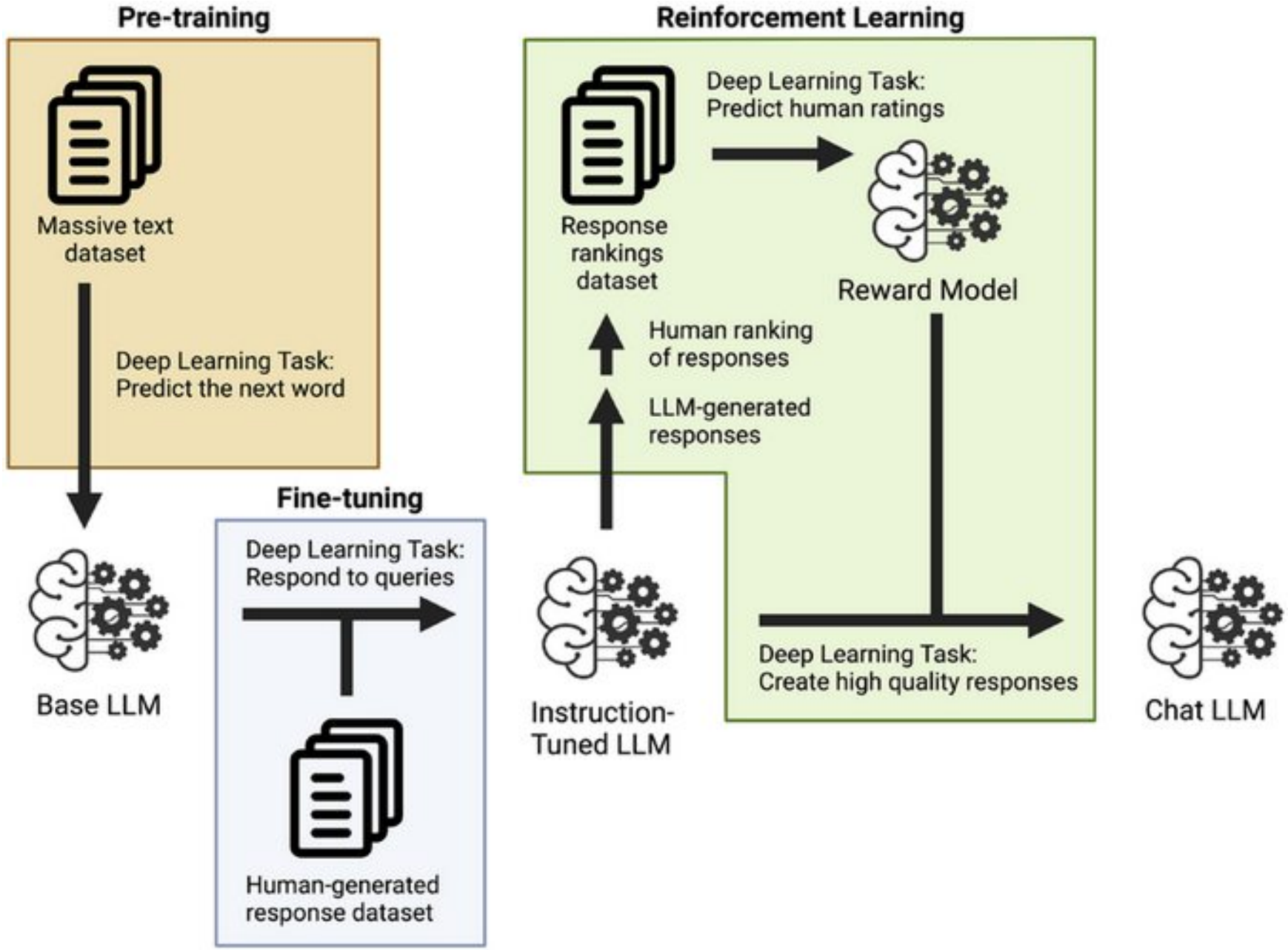
- Recall the main points of LLMs
- Issues and options for improving LLM use without improving LLMs themselves
- Specifically, RAG
- RAG techniques
- Graph RAG

- Another topic: some experiments in reasoning capabilities of LLMs

Recall the main points of LLM on the surface level

What does it do? Predict tokens (letters, words) one at a time:

- John sat in the car and …
- John sat in the car and started …
- John sat in the car and started to …
- John sat in the car and started to drive …
- John sat in the car and started to drive home …


Pre-training
Massive text dataset
Deep Learning Task: Predict the next word
Base LLM
Fine-tuning
Deep Learning Task: Respond to queries
Human-generated response dataset
Reinforcement Learning
Response rankings dataset
Deep Learning Task: Predict human ratings
Reward Model
Human ranking of responses
LLM-generated responses
Instruction-Tuned LLM
Deep Learning Task: Create high quality responses
Chat LLM

Using LLMs

The case of machine learning: gaining new knowledge by machine learning a software system to predict answers or actions similar to what has been observed

Issues:

- A very large number of examples is necessary, as well as a lot of compute
- Errors are inevitable; more generalization leads to more errors
- LLMs: hallucination (errors presented with confidence)
- Inability to explain the reasons for answers given
- Inability to perform search, for example, nontrivial planning or finding proofs
- Inability to efficiently incorporate software tools
- No memory: learning new information is very hard; forgetting is even harder

Issue: no short-term memory, very hard to teach anything, extremely hard to forget

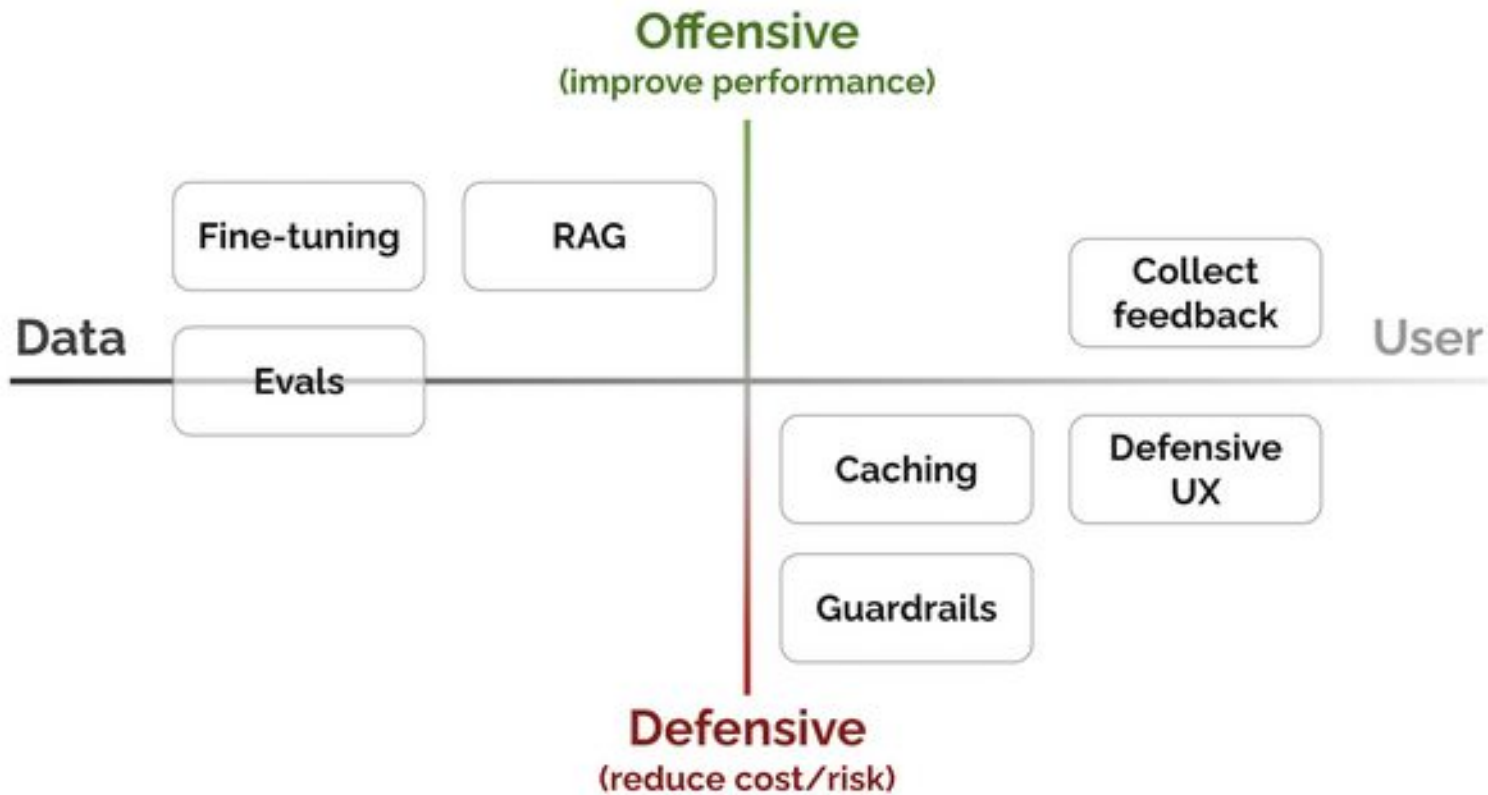
Does not know anything about your specific context / company / organization

Two main approaches to alleviate:

- Retreival-Augmented Generation aka RAG: use special software to find relevant snippets from the docs of your context / organization and add these automatically to the prompt.

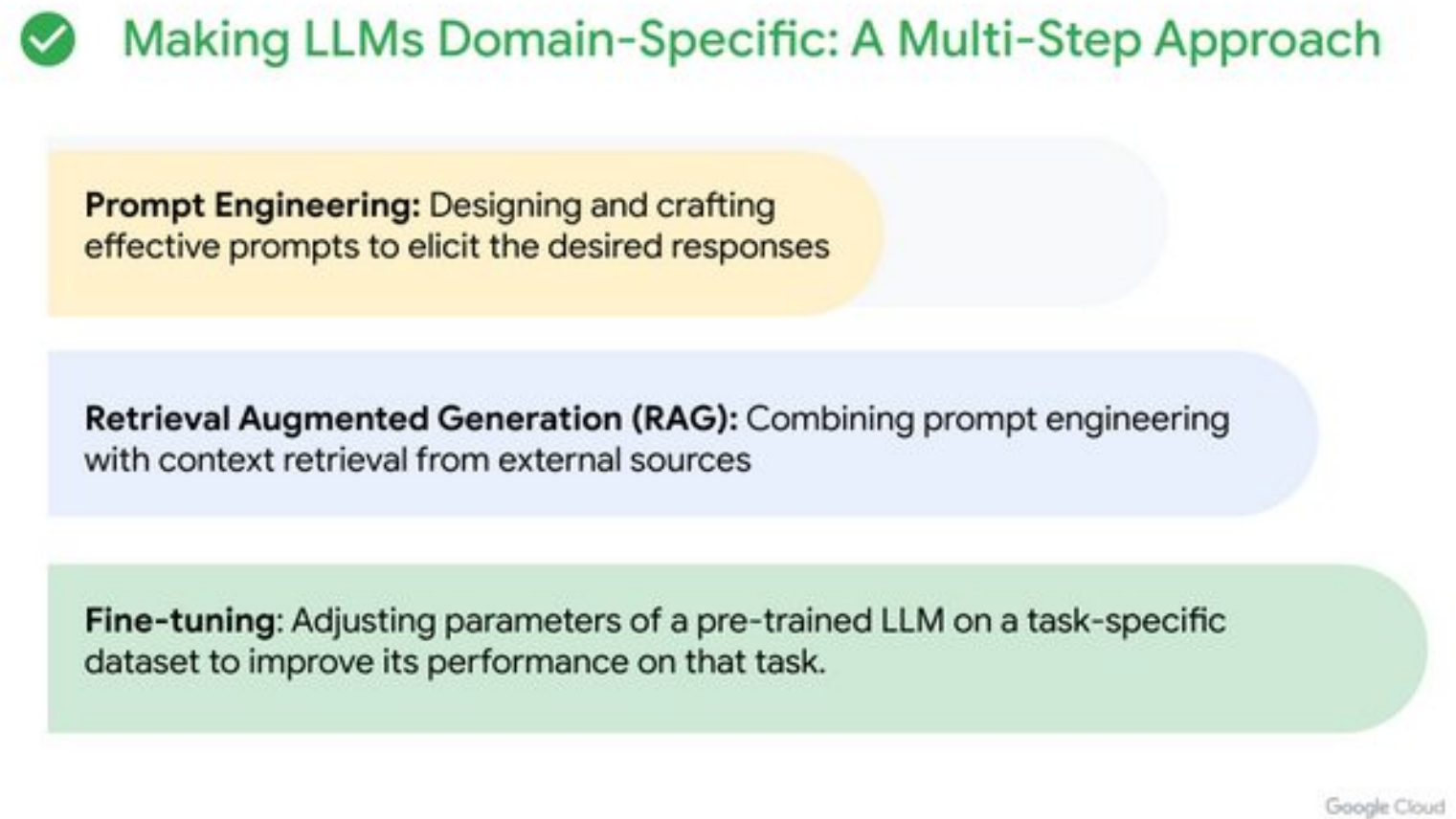
- Fine-tuning: perform additional teaching (like 1000 examples or so) to specialize for your context.

LLM patterns: one view (https://eugeneyan.com/writing/llm-patterns/)



Some Google recommendations:

https://cloud.google.com/blog/products/ai-machine-learning/three-step-design-pattern-for-specializing-llms



Some Google recommendations:

https://cloud.google.com/blog/products/ai-machine-learning/three-step-design-pattern-for-specializing-llms

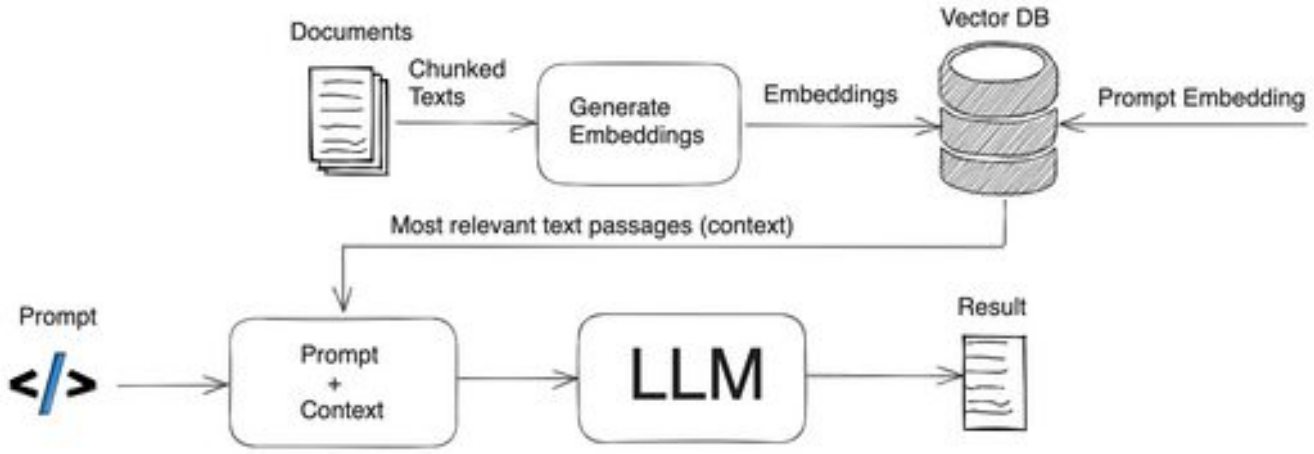
	Prompt Engineering	RAG	Fine-tuning
Key benefit	Rapid adaptability and prototyping	Incorporation of real-time or external data for factual answers	High specialization and tailored responses.
Training requirement	No	No	Yes
External Data	No	Needs a corpus	Task-specific dataset
Computation	No overhead	Overhead for retrieval	Intensive for training, no overhead for inference
Quality Improvement	Iterative refinement	Update/expand corpus	Periodic retraining
Potential Costs	Human labor for crafting prompts	Training, storing corpus, computational overhead	Dataset, training compute, evaluation
Technical Complexity	Low technical	Moderate to high – management of corpus can be complex	Moderate – requires expertise in neural networks & dataset biases
Extra Inference Latency	No	Yes – needed for retrieval	No

RAG (retrieval augmented generation): the main principle of using LLMs in an organization

Idea: search for relevant text snippets and add these to the LLM prompt.

The standard process with specialized RAG software:

- Collect all potentially relevant texts of an application domain and index all their short sections/paragraphs.
- Upon seeing a question, automatically select these sections/paragraphs which seem to be relevant. Use vector similarities / knowledge graphs and final evaluation.
- Automatically add likely relevant text snippets to the question prompt as background.



LLM + RAG based on automatic search

Check out https://perplexity.ai

"When you ask Perplexity a question, it uses advanced AI to search the internet in real time, gathering insights from top-tier sources. It then distills this information into a clear, concise summary, delivering exactly what you need in an easy-to-understand, conversational tone."

- Can find up-to-date sources not used for GPT training
- Experience is like Google mixed with GPT
- Since it uses an LLM (GPT / Gemini / …) it still hallucinates, though less than GPT
- Probably cannot find the documents / materials of your company, but …
- Enterprise version allows uploading your own text, pdfs etc to be used during search for relevant background

Word / phrase vectors for RAG and similar

- Give GPT API a word, sentence or a large piece of text.
- Ask for the vector aka embedding (ca 3000 floating point numbers) for this.
- Suppose you have stored a large number of texts along with vectors.

Then you can, for example, use the vector of the question to find your existing texts / phrases with a similar vector.

LLM patterns: one view (https://eugeneyan.com/writing/llm-patterns/)

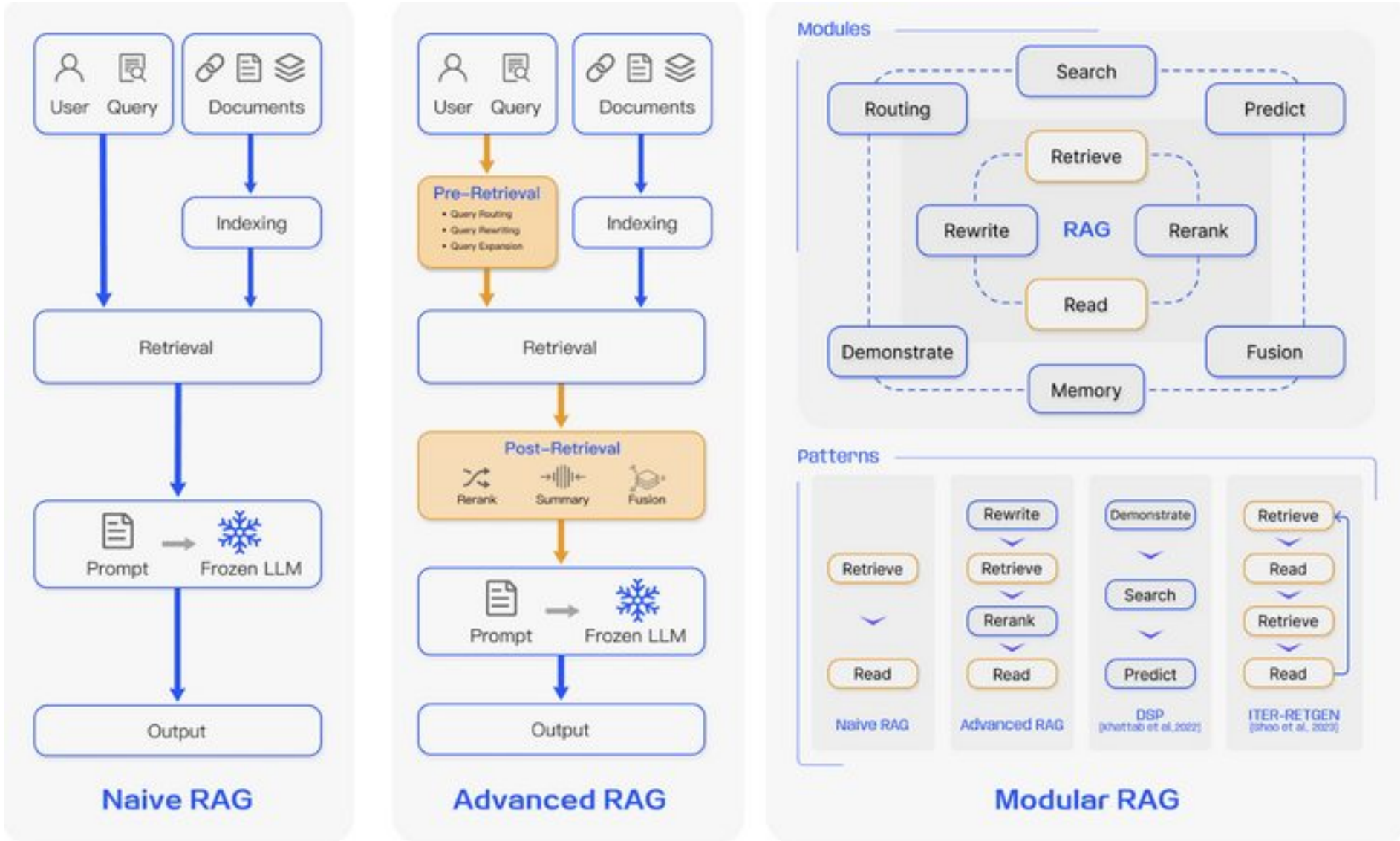
Citation:

Why not embedding-based search only? While it's great in many instances, there are situations where it falls short, such as:

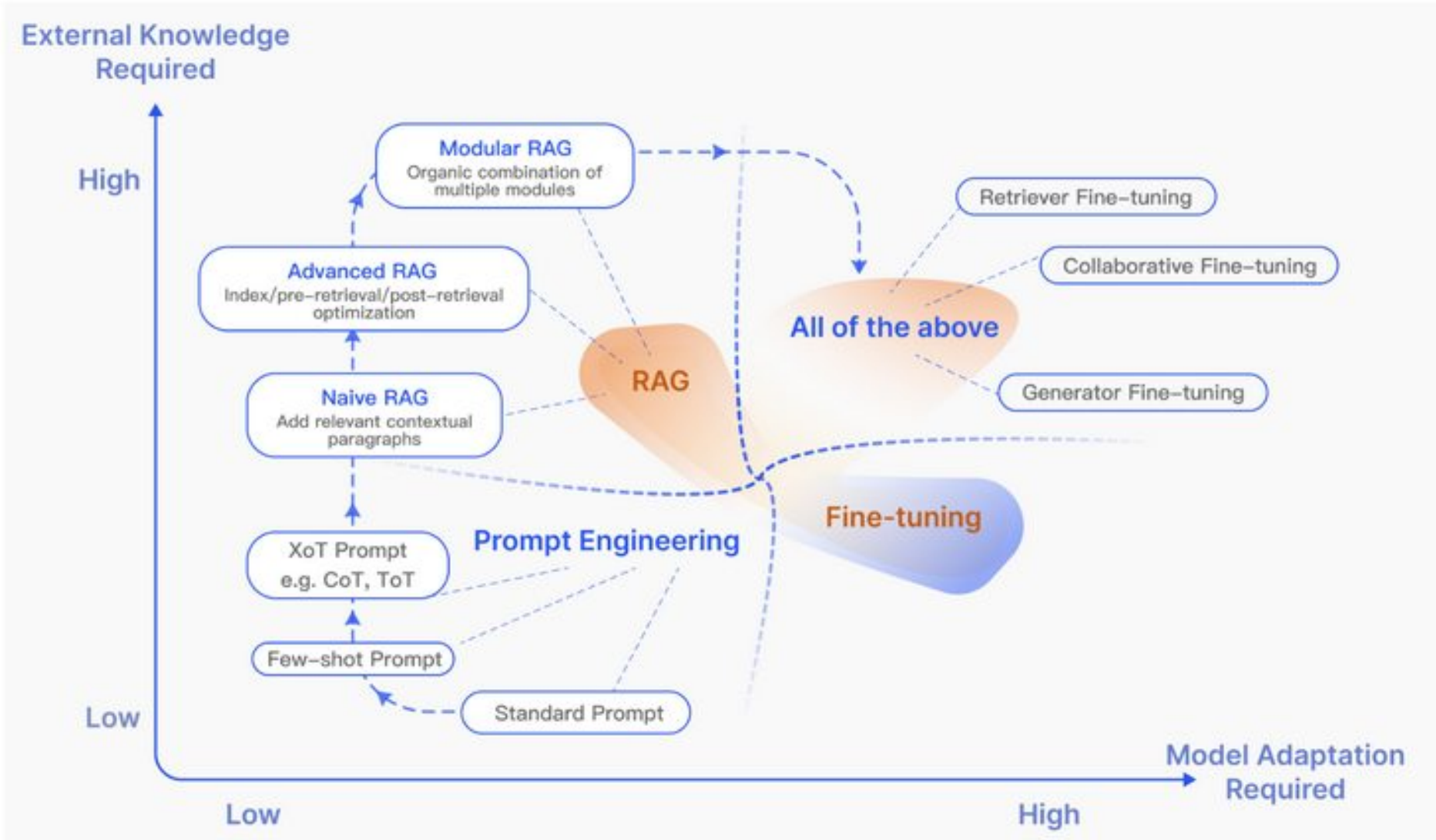
Searching for a person or object's name (e.g., Eugene, Kaptir 2.0)
Searching for an acronym or phrase (e.g., RAG, RLHF)
Searching for an ID (e.g., gpt-3.5-turbo, titan-xlarge-v1.01)

"From experience with Obsidian-Copilot, I've found that hybrid retrieval (traditional search index + embedding-based search) works better than either alone. There, I complemented classical retrieval (BM25 via OpenSearch) with semantic search (e5-small-v2)."

From https://arxiv.org/pdf/2312.10997 (see also https://www.promptingguide.ai/research/rag):

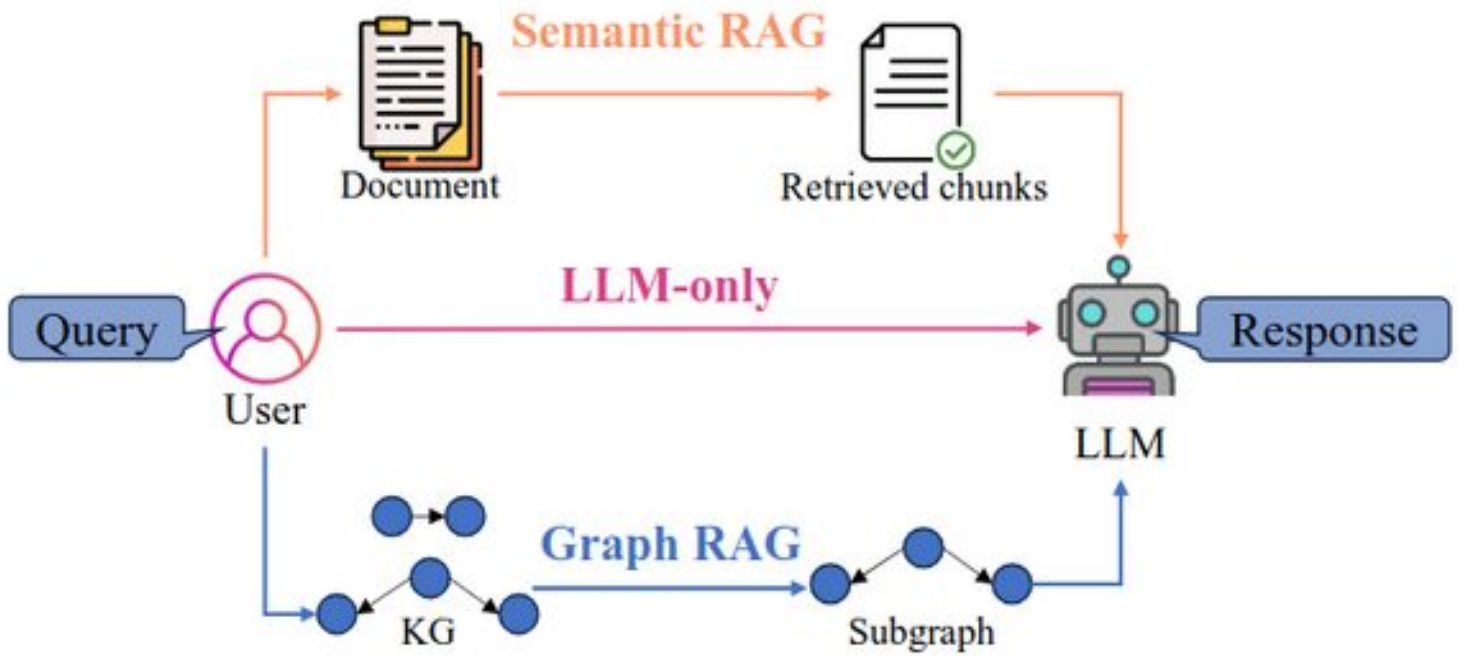


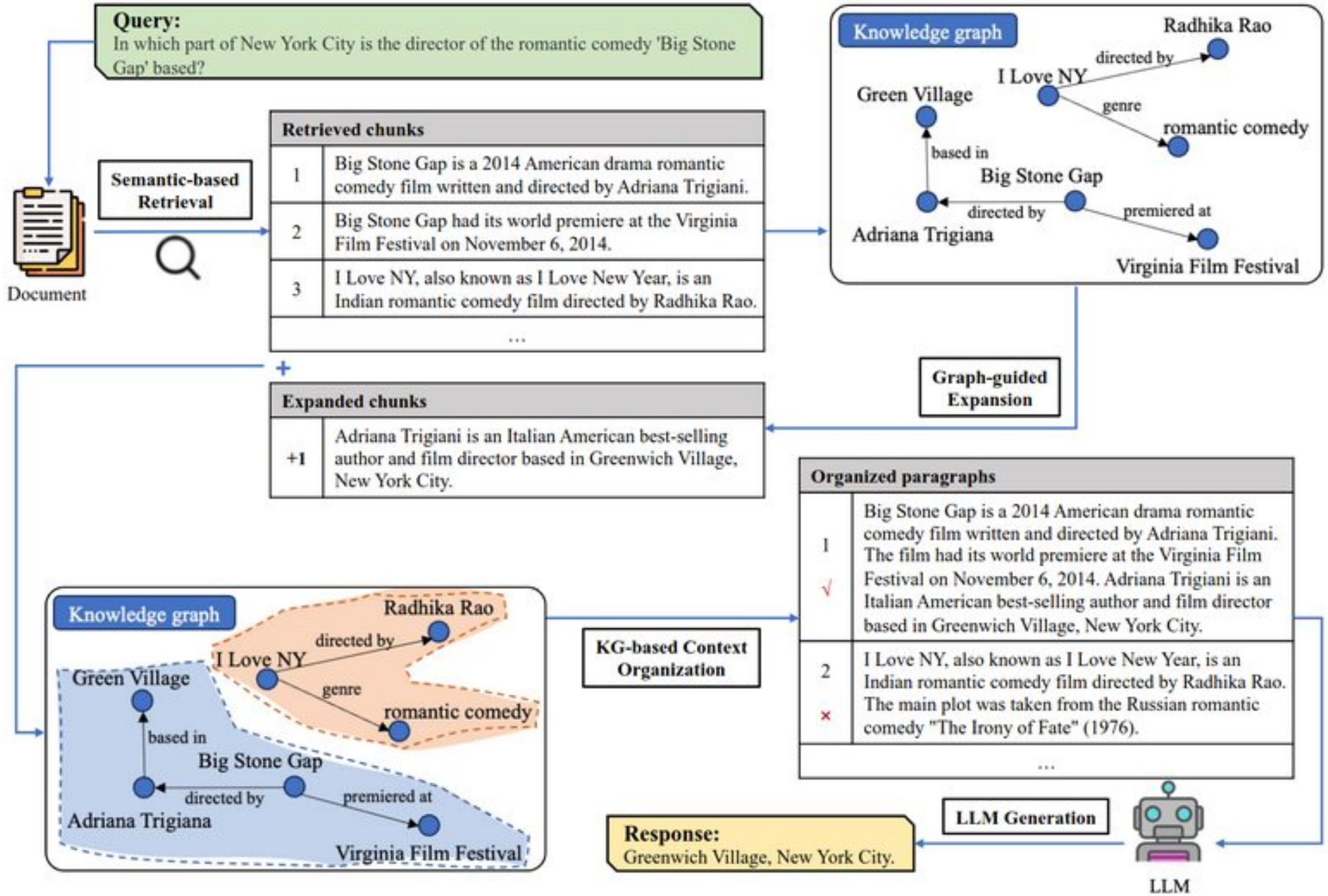
From https://arxiv.org/pdf/2312.10997 (see also https://www.promptingguide.ai/research/rag) :



Graph RAG intro

https://arxiv.org/pdf/2502.06864




Query:
In which part of New York City is the director of the romantic comedy 'Big Stone Gap' based?
Document
Semantic-based Retrieval
Retrieved chunks
1
Big Stone Gap is a 2014 American drama romantic comedy film written and directed by Adriana Trigiani.
2
Big Stone Gap had its world premiere at the Virginia Film Festival on November 6, 2014.
3
I Love NY, also known as I Love New Year, is an Indian romantic comedy film directed by Radhika Rao.
...
Knowledge graph
Radhika Rao
directed by
I Love NY
genre
Green Village
romantic comedy
based in
Big Stone Gap
directed by
premiered at
Adriana Trigiana
Virginia Film Festival
Graph-guided Expansion
+
Expanded chunks
+1
Adriana Trigiani is an Italian American best-selling author and film director based in Greenwich Village, New York City.
Knowledge graph
Radhika Rao
directed by
I Love NY
genre
Green Village
romantic comedy
based in
Big Stone Gap
directed by
premiered at
Adriana Trigiana
Virginia Film Festival
KG-based Context Organization
Organized paragraphs
1
√
Big Stone Gap is a 2014 American drama romantic comedy film written and directed by Adriana Trigiani. The film had its world premiere at the Virginia Film Festival on November 6, 2014. Adriana Trigiani is an Italian American best-selling author and film director based in Greenwich Village, New York City.
2
×
I Love NY, also known as I Love New Year, is an Indian romantic comedy film directed by Radhika Rao. The main plot was taken from the Russian romantic comedy "The Irony of Fate" (1976).
...
LLM Generation
LLM
Response:
Greenwich Village, New York City.

Prompt for Triplet Extraction

Instruction:
Extract informative triplets directly from the text following the examples. Do not add any extra words, line breaks, or spaces.
Example 1:
Text: Scott Derrickson (born July 16, 1966) is an American director, screenwriter and producer.
Triplets:
<Scott Derrickson, born in, 1966>,
<Scott Derrickson, nationality, America>,
<Scott Derrickson, occupation, director>,
<Scott Derrickson, occupation, screenwriter>,
<Scott Derrickson, occupation, producer>
Example 2:
Text: A Kiss for Corliss is a 1949 American comedy film directed by Richard Wallace and written by Howard Dimsdale.
Triplets:
<A Kiss for Corliss, year, 1949>,
<A Kiss for Corliss, country, America>,
<A Kiss for Corliss, genre, comedy film>,
<A Kiss for Corliss, director, Richard Wallace>,
<A Kiss for Corliss, writer, Howard Dimsdale>

Target Text: <target text>
Triplets:

An additional short Graph RAG intro

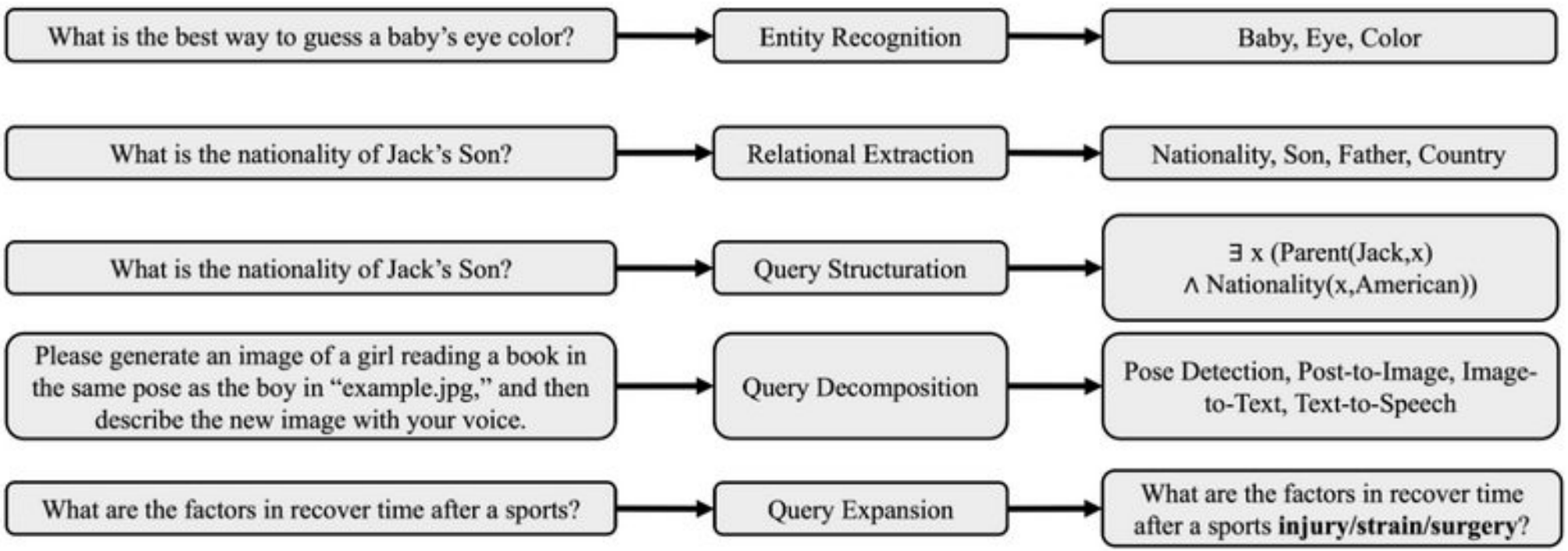
https://medium.com/ai-agent-insider/what-is-graph-rag-a-key-benefit-of-graphrag-aa99cff02ae3

Cite: "… using knowledge graphs to store and retrieve structured information. Unlike traditional RAG, which relies only on vector similarity search, Graph RAG:

- Stores data as nodes (entities) and edges (relationships).
- Enables graph traversal to fetch related concepts.
- Combines vector similarity with graph-based reasoning.
- Provides explainable and structured responses."

GraphRag intro and tools from Microsoft:

https://arxiv.org/pdf/2501.00309



GraphRag intro and tools from Microsoft:

https://arxiv.org/pdf/2404.16130

https://www.microsoft.com/en-us/research/blog/graphrag-unlocking-llm-discovery-on-narrative-private-data/

https://microsoft.github.io/graphrag/

Citation: “GraphRAG, creates a knowledge graph based on an input corpus. This graph, along with community summaries and graph machine learning outputs, are used to augment prompts at query time”

Routers and GraphRag
https://ragaboutit.com/building-a-graph-rag-system-with-llm-router-a-comprehensive-coding-walkthrough/

Fine-tune (or N-shot prompt ask) to detect a suitable method (LLM selection / prompt / RAG) for a query type:

```
routing_data = [
    {"query": "What is the capital of France?", "route": "simple_qa"},
    {"query": "Explain the theory of relativity", "route": "complex_explanation"},
    {"query": "Generate a poem about autumn", "route": "creative_task"},
    {"query": "Summarize the latest research on quantum computing", "route": "research_summary"}
]
```

Reasoning: partially OK

- LLMs perform limited reasoning.
- Nobody knows where exactly is the plateau or limit.
- Deep / uncommon reasoning fails, though.
- LLMs do not contain search / planning / simulation algorithms.

Recent paper by Apple research: “adding seemingly relevant but ultimately inconsequential information to the logical reasoning of the problem led to substantial performance drops of up to 65% across all state-of-the-art models. “